

INTERACTION DESIGN

Cambridge Computer Science Tripos – Part IA
Paper 3 – Complete Revision Guide

Based on past Tripos papers 2019–2025 & course lecture material

Contents

1. User-Centred Design (UCD) & the Design Process

User-Centred Design places the end-user at the heart of every stage of design and development. It is iterative – you cycle through stages repeatedly rather than moving linearly.

1.1 The Four-Stage Iterative Cycle

The standard UCD cycle has four phases that repeat until the product is satisfactory:

1. Identify needs and establish requirements – understand who the users are, what they need, and what constraints exist.
2. Design alternatives – generate possible solutions (conceptual and physical design).
3. Prototype – build a representation of the design at an appropriate fidelity for testing.
4. Evaluate – assess how well the design meets user needs; findings feed back into stage 1.

⚠ Exam Alert – 2019, 2020 papers

- The practical sessions focused on stages 1 (requirements), 2 (design), and 3 (prototyping).
- They did NOT focus on full evaluation with real end-users or deployment/maintenance.
- This means usability issues may remain undetected; the app may not fully meet all user needs.

1.2 Why Iterative?

- Designs are rarely right first time – early prototypes reveal unforeseen problems.
- Cheap to fix mistakes early (paper prototype) vs. late (shipped product).
- User needs evolve; iteration keeps the design aligned.

1.3 What UCD Achieved in Practice (Weather App / Career App examples)

Stages covered: requirements gathering (interviews, observation, questionnaires), lo-fi prototyping, hi-fi implementation. Stages not covered: full summative evaluation with diverse real users, long-term deployment, accessibility testing with real target populations.

2. Stakeholders & Requirements Analysis

2.1 Stakeholder Analysis – CUSTOM Framework

CUSTOM identifies four categories of stakeholder relevant to a system:

Category	Who they are & example (Weather App)
C – Clients / Customers	Organisations or individuals who commission or pay for the system. E.g. a company commissioning the weather app.
U – Users (primary)	People who directly interact with the system regularly. E.g. event organisers who rely on the app daily.
S – Secondary stakeholders	People who benefit from or are affected by the system but do not use it directly. E.g. event attendees who experience decisions made using the app.
T – Tertiary / Other	Wider community, regulatory bodies, those affected indirectly. E.g. health & safety authorities; data protection regulators.

Note: 'CUSTOM' variations in the literature sometimes include 'O' (Other) and 'M' (Maintenance) – be guided by what your course notes say. The key is to show you can distinguish primary users from indirect stakeholders.

2.2 Stakeholder Analysis for ChatGPT (2024 paper)

Identify four distinct groups with rationale:

- End users (individuals) – interact directly with the chatbot to get answers, compose text, etc.
- Developers / OpenAI engineers – build and maintain the model; affected by user feedback.
- Enterprise clients – companies that integrate ChatGPT via API into their own products.
- Regulators / policymakers – concerned with safety, data privacy, and AI legislation.

2.3 Personas

A persona is a fictional but research-based representation of a typical user. Each persona has:

- Name, age, job, and background.
- Goals, motivations, and frustrations.
- Technology habits and context of use.

Use personas to make design decisions concrete: 'Would Maya, our 19-year-old student persona, understand this menu label?'

- ✓ **Exam tip – Personas (2024 q6, 2025 q5)**
- ✓ Create TWO distinct personas that represent different user segments.
- ✓ Show how the personas lead to different or complementary design choices.

- ✓ Give each persona a specific pain point the app addresses.

2.4 Requirements Analysis

Requirements Analysis

The process of identifying, organising, and prioritising what a system must do (functional requirements) and how well it must do it (non-functional requirements), based on data gathered from users and stakeholders.

Functional requirements – what the system must do. Example: 'The app shall allow users to view today's weather for a specified city.'

Non-functional requirements – quality attributes. Example: 'The app shall load within 2 seconds on a standard mobile connection.' Covers: performance, accessibility, security, usability, scalability.

⚠ Common exam requirement for ID questions

- Always justify each requirement – say WHY it is needed for your chosen user group.
- Use 'must', 'shall', or 'should' to signal requirement strength.
- Include at least one accessibility or inclusion requirement when the user group has specific needs.

3. Data Gathering Methods

Data gathering informs requirements and evaluates designs. Choose methods that suit your user group, context, and research questions.

3.1 Overview of Methods

Method	What it is	When to use / pros & cons
Interviews (structured, semi-structured, unstructured)	Direct conversation with users to probe attitudes, experiences, and needs.	Rich qualitative data; good for exploring unknown territory. Time-intensive; small sample.
Questionnaires / Surveys	Written questions distributed to many participants.	Large sample, quantifiable, quick. Less depth; low response rates; can't probe.
Observation (naturalistic / structured)	Watching users in their real environment.	Reveals what users actually do vs. what they say they do. Can be intrusive; time-consuming.
Focus Groups	Moderated group discussion of 5–10 users.	Generates debate and range of views. Group dynamics can skew results; dominant voices.
Diary Studies	Participants record their own experiences over time.	Captures longitudinal behaviour. Relies on participant compliance.
Contextual Inquiry	Interview + observation in the user's natural context.	Deeply contextual; reveals tacit knowledge. Labour-intensive.

3.2 Difficult Methods for Specific Populations

For visually impaired users (2020, 2021 papers):

- Questionnaires with visual layout or Likert scales are problematic if the questionnaire itself is not accessible.
- Observation of facial expressions or gestures is not possible.
- Card sorting (physical) requires adaptation.
- Appropriate methods: interviews, think-aloud (verbal), contextual inquiry, diary studies.

For a mindfulness app targeting students with anxiety (2023 paper):

- Difficult: observation in therapy/mindfulness sessions (sensitive context); long interviews may be stressful.
- Appropriate: diary studies (low-effort, reflective), questionnaires on stress/anxiety scales, short semi-structured interviews.

3.3 Analysing Interview Data

Typical process after collecting interviews:

5. Transcribe recordings.
6. Open coding – label segments with descriptive codes.

7. Thematic analysis – group codes into themes.
8. Affinity diagramming – physically group findings on sticky notes.
9. Triangulate with other data sources to improve validity.

Validity: use multiple methods (triangulation), member-checking, or peer review. Reliability: use multiple coders; measure inter-rater agreement.

4. Task Analysis

Task Analysis

Decomposing tasks into a sequence of sub-tasks and actions to understand how users achieve their goals. It models what users DO, not what the system does.

Most common form: Hierarchical Task Analysis (HTA) – a tree of goals, sub-goals, and operations, with plans specifying when to do each.

4.1 HTA Example – 'Find today's temperature in city X' (Weather App not yet open)

- 0. Find today's temperature in city X
 - 1. Open Weather App
 - 1.1 Locate app icon on desktop
 - 1.2 Double-click to launch
 - 2. Enter city name
 - 2.1 Click search/city field
 - 2.2 Type city name
 - 2.3 Submit search
 - 3. Read temperature from display

Plan 0: Do 1, then 2, then 3. Plan 2: Do 2.1, 2.2, then 2.3.

4.2 Limitations of Task Analysis

- Assumes users follow a fixed sequence – ignores opportunistic, exploratory behaviour.
- Doesn't capture cognitive load or emotional states.
- Difficult to apply when tasks are highly variable or context-dependent.
- Can miss 'tacit knowledge' – things experts do automatically without conscious thought.
- Doesn't model errors or recovery paths well.

5. Prototyping

Prototyping

Creating a preliminary version of a product to explore design ideas and test them with users before full implementation. Prototypes exist on a fidelity spectrum from rough sketches to near-final products.

5.1 Lo-Fidelity Prototypes

- Paper prototypes, storyboards, wireframes on paper.
- Quick and cheap to create and modify.
- Used early to test overall structure and navigation, not visual detail.
- Encourage user feedback at an early stage without intimidating users.

5.2 Hi-Fidelity Prototypes

- Digital mockups (e.g. Figma, Adobe XD) or working software prototypes.
- Look and feel close to the final product.
- Used for detailed usability testing and stakeholder sign-off.
- More expensive to change.

5.3 Prototyping Methods

Method	Description
Paper prototyping	Hand-drawn sketches of screens; a 'wizard of Oz' facilitator simulates system responses. Fast, cheap, very early stage.
Wireframing	Low-detail digital layouts showing structure without colour/imagery. Tools: Balsamiq, Figma wireframe mode.
Mockup / Hi-fi digital	High-detail static screens showing final look-and-feel. Tools: Figma, Sketch, Adobe XD.
Interactive prototype	Clickable prototype linking screens together to simulate navigation. Still no back-end logic.
Vertical prototype	Full implementation of a small number of features to test depth.
Horizontal prototype	Shallow implementation across all features to test breadth/coverage.

- ✓ **Exam tip – Prototyping (2023 q6, 2024 q6, 2025 q5)**
- ✓ When asked to sketch screens: draw 2–4 screens with labels and captions.
- ✓ Show how the screens link to your stated requirements.
- ✓ Include interactive elements (buttons, text fields, navigation) with descriptions of what happens.
- ✓ For 'prototyping methods': name the method, say why it's appropriate for THIS project, and what it enables you to test.

6. Information Architecture (IA)

Information Architecture

The structural design of shared information environments. In interaction design it refers to how content and functionality are organised, labelled, and made navigable within a system.

Key components of IA:

- Organisation systems – how content is categorised (hierarchical, faceted, sequential).
- Labelling systems – names given to navigation items, pages, and sections.
- Navigation systems – menus, breadcrumbs, search, links.
- Search systems – how users find content directly.

6.1 IA Diagram

An IA diagram (also called a site map) shows the hierarchical structure of a system as a tree. Each node represents a screen/page/section; lines show navigation relationships.

Example for a Career App (2025 paper):

- Home Dashboard
 - Resume Builder
 - Edit sections (Summary, Experience, Education, Skills)
 - Preview & Download
 - Interview Prep
 - Question Bank
 - Mock Interview Mode
 - Job Search
 - Browse Listings
 - Saved Jobs
 - Networking Tips
 - Profile
 - Settings & Notifications

6.2 IA Usability Issues

Common IA problems spotted in exam questions (e.g. CamCORS 2023):

- Ambiguous labels – users don't know which section to choose.
- Too many top-level categories – increases cognitive load.
- Important functions buried too deep in hierarchy.
- Inconsistent labelling (same thing called different names on different pages).

7. Nielsen's 10 Heuristics & Heuristic Evaluation (HE)

Heuristic Evaluation

An expert-based usability inspection method in which evaluators examine an interface and judge its compliance with a set of usability principles (heuristics). Typically 3–5 evaluators; each works independently, then findings are combined.

7.1 The 10 Heuristics – Full Reference Table

# Heuristic	Description & what to look for
1. Visibility of System Status	System should always keep users informed about what is happening (e.g. loading indicators, progress bars, confirmation messages). Violation: no feedback after clicking a button.
2. Match between System and Real World	Use words, concepts, and metaphors familiar to the user rather than system-oriented jargon. Violation: technical error codes shown to end users.
3. User Control and Freedom	Support undo/redo and easy exits so users can recover from mistakes. Violation: no undo option after deleting all chats.
4. Consistency and Standards	Follow platform conventions and be consistent within the product. Violation: same action labelled differently on different screens.
5. Error Prevention	Design to prevent errors before they occur (e.g. confirmations, disabling irrelevant options). Violation: no confirmation before deleting all chats.
6. Recognition rather than Recall	Minimise memory load by making options, objects, and actions visible. Violation: requiring users to remember commands; no visible hint.
7. Flexibility and Efficiency of Use	Provide shortcuts for expert users while remaining accessible for novices. Violation: no keyboard shortcuts; everything requires clicking through menus.
8. Aesthetic and Minimalist Design	Do not include irrelevant or rarely needed information. Clutter increases cognitive load. Violation: overwhelming first-run screen with too many options.
9. Help Users Recognise, Diagnose, and Recover from Errors	Error messages should be plain language, identify the problem precisely, and suggest a solution. Violation: 'Error 404' with no explanation or next step.
10. Help and Documentation	Provide easy-to-search, task-focused help. Violation: no help section; or help only accessible via a hidden link.

7.2 How to Conduct Heuristic Evaluation

10. Brief evaluators on the heuristics and the context of use.
11. Each evaluator independently inspects the interface.
12. Record each violation: which heuristic, where it occurs, severity rating (0–4).
13. Aggregate findings across evaluators; combine and de-duplicate.
14. Prioritise issues by severity for redesign.

7.3 Severity Ratings

Severity	Meaning
0	Not a usability problem at all.
1	Cosmetic only; fix if time allows.
2	Minor usability problem; low priority fix.
3	Major usability problem; important to fix.
4	Usability catastrophe; must fix before release.

7.4 HE vs Cognitive Walkthrough – When to Use Which

Criterion	Heuristic Evaluation vs Cognitive Walkthrough
Focus	HE: broad usability audit across entire interface. CW: focused on specific tasks and learnability.
Expertise needed	HE: usability experts who know the heuristics. CW: can be done by evaluators who understand user goals.
Output	HE: list of violations with heuristic labels. CW: step-by-step analysis of task success/failure.
Novel / unusual systems	CW more appropriate when system has no obvious convention to compare against (e.g. colour-to-sound device – 2020 paper).
General usability survey	HE more appropriate when you want broad coverage quickly.

7.5 HE on Microsoft Teams (2025 paper) – Model Answers

Problem	Heuristic violated & reason
(i) Buffering/long page loading on internet interruption	1 – Visibility of System Status. No clear feedback to user about connection state or loading progress.
(ii) No demonstration of common tasks	10 – Help and Documentation. Users need documentation or onboarding demonstrations; absence leaves novices unsupported.
(iii) No wizard to guide starting an activity	6 – Recognition rather than Recall (also arguably 10). Users must remember how to start; no step-by-step guidance.
(iv) Layout not in order of steps to take	2 – Match between System and Real World. The real-world workflow is not reflected in the UI layout.
(v) Difficult for untrained users due to feature complexity	7 – Flexibility and Efficiency of Use. System caters to experts but not novices; no progressive disclosure.
(vi) No advanced search feature	7 – Flexibility and Efficiency of Use. Expert users lack shortcuts; all users must browse rather than search precisely.
(vii) Posts overwrite important files/assignments	8 – Aesthetic and Minimalist Design (also 1). Important information is buried or overwritten; interface lacks hierarchy.
(viii) Search leads to parent	4 – Consistency and Standards. Inconsistent with user expectation that

Problem	Heuristic violated & reason
'Microsoft' not Teams directly	searching 'Teams' yields the Teams app.
(ix) Navigation on left, hard to reach on mobile	2 – Match between System and Real World / platform conventions. Mobile convention places navigation at bottom; left nav is a desktop pattern.

7.6 Trade-offs Between Heuristics

Asked whether fewer heuristics are better (2019 paper):

- Combining heuristics makes evaluation faster but can cause evaluators to miss violations.
- E.g. heuristics 1 (visibility) and 9 (error recovery) could be combined under 'feedback', but this may miss subtle visibility issues that aren't errors.
- More heuristics = more thorough but harder to learn and apply consistently.
- Trade-off: completeness vs. practicality and reliability of application.

8. Cognitive Walkthrough (CW)

Cognitive Walkthrough

An inspection method that simulates a user's problem-solving process for each step of a task to identify learnability issues. Based on the theory that novice users learn by exploration.

8.1 The Four Questions

For each action in the task sequence, evaluators ask:

15. Will users try to achieve the right effect? (Do users know what to do at this step?)
16. Will users notice that the correct action is available? (Is the control visible and recognisable?)
17. Will users associate the action with the desired effect? (Is the label/icon meaningful?)
18. If the action is correct, will users see that progress is being made? (Is there adequate feedback?)

8.2 How to Conduct a CW

19. Define the user population and their knowledge level.
20. Choose a representative task.
21. Decompose the task into a sequence of actions (like an HTA).
22. For each action, answer the four questions; record successes (S) and failures (F).
23. Failures indicate usability problems; document them.
24. Produce a list of design issues and suggestions.

8.3 Example – Weather App CW (2021 paper)

Task: (from main screen) View detailed weather forecast for Field Day.

Action	Analysis
1. Locate 'Field Day' event in the main screen list.	Q1 OK – user knows they need to find the event. Q2 Potential fail – is the event clearly visible? Is there enough contrast?
2. Tap on 'Field Day' to open event details.	Q3 Potential fail – it may not be obvious that tapping the event opens it (no affordance cue like a chevron). Q4 OK if a new screen loads with a transition.
3. Find the 'Weather Forecast' section within event details.	Q2 Potential fail – if forecast is not prominently labelled or is below the fold. Q3 Potential fail – label 'weather' vs. 'forecast' ambiguity.
4. Read detailed forecast data.	Q4 OK once data is visible.

8.4 CW Instruction Sheet Format (2022 paper style)

When writing an instruction sheet for first-time CW evaluators, include:

- Brief definition of CW and its purpose (learnability focus).

- Description of the user persona (e.g. business manager, first-time user).
- The specific task to evaluate (e.g. 'Find the latest monthly sales total for France').
- The decomposed action sequence step by step.
- The four questions to ask at each step.
- What to record: success/failure for each question, description of the issue, potential cause.

8.5 Generating Redesign Suggestions from CW

After a CW, translate each failure into a specific design suggestion. Example failures → fixes:

- No affordance cue on tappable items → Add chevron icon or underline to indicate interactivity.
- Ambiguous label 'Weather' → Rename to 'Detailed Forecast' and add a weather icon.
- No loading feedback → Add a progress spinner while data loads.
- Volume setting allows impossible value (150) → Add input validation with max value of 100.

9. Gestalt Theory & Principles

Gestalt Theory

A psychological theory that describes how humans perceive visual elements as organised wholes rather than individual parts. The key insight is that 'the whole is more than the sum of its parts.' For interaction design, Gestalt principles explain how layout and grouping affect perception and usability.

9.1 The Six Core Gestalt Principles

Principle	Definition & Interaction Design Application
Proximity	Elements placed close together are perceived as belonging to the same group. Use: group related form fields; place labels next to their inputs; keep related nav items together.
Similarity	Elements that look alike (same colour, size, shape, or texture) are perceived as related. Use: use consistent button styles for all primary actions; use same icon style for related functions.
Continuity (Good Continuation)	The eye follows smooth, continuous paths. Elements arranged along a line or curve are perceived as related. Use: align elements along a grid; use visual flows to guide users through a form.
Closure	People perceive incomplete shapes as complete by filling in gaps. Use: progress indicators (a partial circle suggests completeness); card borders; incomplete shapes used as icons.
Figure/Ground	People separate visual fields into a 'figure' (focus) and 'ground' (background). Use: modals/overlays that dim the background; contrast between content and background; card elevations.
Common Fate	Elements moving in the same direction are perceived as a group. Use: animated transitions; items that slide in together; parallax effects on related content.

9.2 Applying Gestalt to Interface Analysis

When a question shows you an interface and asks you to identify Gestalt principles, use this method:

25. Scan for visual groupings (proximity) – are related items near each other?
26. Look for visual similarities (colour, shape, size) that group elements.
27. Identify lines, alignment, and paths the eye follows (continuity).
28. Note any borders, shadows, or negative space creating figure/ground separation.
29. State the principle, where it is applied, and what effect it has on the user.

⚠ Exam Alert – Gestalt (2019 q6, 2020 q6, 2022 q5, 2024 q6)

- Almost every year asks about Gestalt. Memorise all six principles with one clear example each.
- When applying to a dashboard (2022): proximity groups countries; similarity groups data types; continuity in the bar chart baseline.
- When asked to APPLY Gestalt to your own design: show how each principle is

deliberately used, not accidental.

10. Human Memory & Attention in Interaction Design

10.1 Human Memory

Types of Memory

Memory Type	Relevance to ID
Working memory (short-term)	Capacity $\sim 7 \pm 2$ items (Miller's Law). Interfaces should minimise the number of things a user must hold in mind simultaneously. Chunking information aids working memory.
Long-term memory	Stores schemas and learned behaviours. Users bring mental models from past experience. Design should match existing mental models.
Recognition vs. Recall	Recognition (seeing something and knowing it) is easier than recall (generating from memory). Good design favours recognition: visible menus, icons with labels, autocomplete. This corresponds to Heuristic 6.

Design Implications

- Don't require users to remember information from one screen to use it on the next.
- Use consistent terminology – reduces learning load.
- Progress indicators and breadcrumbs reduce need to remember where you are.
- Autocomplete and defaults help users recognise options rather than recall them.

10.2 Human Attention

Types of Attention

Attention Type	Relevance to ID
Selective attention	Users focus on one thing at a time; other things fade. Important content must be salient (high contrast, bold, movement).
Divided attention	Performing two tasks simultaneously is difficult. Avoid requiring users to switch attention between many UI areas at once.
Sustained attention (vigilance)	Attention degrades over time. Long forms, multi-step processes, and monitoring tasks are cognitively taxing.
Pre-attentive processing	Certain features (colour, size, shape, motion) are processed before conscious attention. Use for visual hierarchy and urgent alerts.

Design Implications

- Use visual hierarchy (size, weight, colour) to direct attention to the most important elements first.
- Avoid animation on non-critical elements – it hijacks selective attention.
- Use pre-attentive cues (red, blinking icons) only for urgent notifications.
- Microsoft Teams (2025 paper): notification badges on the activity icon use pre-attentive colour/number to draw attention. But if every icon has a badge, the hierarchy collapses.

11. Card Sorting & Similarity Analysis

Card Sorting

A user research technique in which participants sort cards (each labelled with a piece of content or functionality) into groups that make sense to them. Used to discover users' mental models for organising information, informing IA.

11.1 Open vs Closed Card Sorting

Type	Description
Open card sort	Participants create their own group names. Reveals how users categorise; informs new IA.
Closed card sort	Groups are predefined; participants assign cards. Tests an existing IA structure.

11.2 Similarity Analysis

After open card sorting, similarity analysis produces a numerical measure of how often each pair of cards was placed in the same group across participants.

Similarity Rating formula:

Similarity(A, B) = (number of users who placed A and B in the same group) / (total number of users) × 100%

Result is a similarity matrix. High similarity (close to 100%) means most users group those cards together – they should be co-located in the IA.

11.3 Worked Example (2019 Paper)

4 cards (1,2,3,4), 3 users (A,B,C). Groups:

- User A: {1,4}, {2,4}, {1} – wait, checking: groups are {1,4}, {2,4}, {1}. Re-reading: User A groups are listed as '1,4' and '2,4' and '1'. The table shows three separate groupings; each row = one group. Let's work with the data as given.

From the exam data as given – calculate all pairs:

Pair	Users who co-grouped → Similarity
(1,2)	Check each user: A: {1,4},{2,4},{1} – 1 and 2 not together. B: {1,2,3},{1,2,4},{3,4} – yes in {1,2,3} and {1,2,4}. C: {1,3},{4},{3,4} – no. → 1/3 ≈ 33%
(1,3)	A: no. B: yes in {1,2,3}. C: yes in {1,3}. → 2/3 ≈ 67%
(1,4)	A: yes in {1,4}. B: yes in {1,2,4}. C: no ({4} alone). → 2/3 ≈ 67%
(2,3)	A: no. B: yes in {1,2,3}. C: yes in {3,4}? no – {1,3} and {4} and {3,4}: 2&3 not together. → 1/3 ≈ 33%
(2,4)	A: yes in {2,4}. B: yes in {1,2,4}. C: no. → 2/3 ≈ 67%

Pair	Users who co-grouped → Similarity
(3,4)	A: no. B: yes in {3,4}. C: yes in {3,4}. → $2/3 \approx 67\%$

Commentary: Cards 1&3, 1&4, 2&4, 3&4 all share 67% similarity, suggesting they are moderately associated. No pair reaches 100% consensus, indicating users have varied mental models. The IA design might group {1,3,4} and {2,4} as overlapping clusters, or further user testing is needed to resolve ambiguity.

- ✓ **Exam tip – Card sorting**
- ✓ Show your working step by step for each pair.
- ✓ Always comment on what the results imply for the IA: which cards to co-locate, and where agreement is weak.
- ✓ Note when sample size is too small ($n=3$) to draw strong conclusions.

12. Design Principles

12.1 Core Usability Goals

Goal	Definition
Effectiveness	Can users complete their goals correctly and completely?
Efficiency	Can users achieve goals with minimal effort/time?
Safety	Can users avoid and recover from errors?
Utility	Does the system provide the right functions for what users need to do?
Learnability	How easy is it for new users to learn to use the system?
Memorability	How easily can infrequent users remember how to use it after a period of absence?

12.2 User Experience (UX) Goals

Beyond usability, UX goals address subjective experience: satisfying, enjoyable, fun, entertaining, helpful, motivating, aesthetically pleasing, creative, rewarding. These are harder to measure but increasingly important.

12.3 Product-Specific Design Principles

A product-specific design principle is a custom guideline derived from the specific context, user group, and requirements of the product. Examples from past papers:

App	Example Product-Specific Principle
COVID-19 test app (2022)	'Clarity under stress': because users may be anxious or unwell, all notifications and instructions must use plain language with large text, single-action screens, and no ambiguity.
Mindfulness app (2023)	'Non-judgmental presentation': the interface must never use language that could trigger shame or comparison; progress should be framed positively.
Financial literacy app (2024)	'Trust through transparency': all financial calculations must show how they are derived so users build genuine financial understanding rather than dependency on the app.
Career app (2025)	'Progressive empowerment': the app should start simple and gradually reveal more advanced features as users build confidence.

- ✓ **Exam tip – Design principles**
- ✓ Don't just state a general HCI principle (e.g. 'be consistent'). **MOTIVATE** it: explain why this specific product needs this specific principle given the user group.
- ✓ Link the principle to specific stakeholder needs or requirements.

12.4 Usability vs Privacy/Security Trade-offs

Classic trade-offs (2022 paper):

- Login friction: strong authentication (2FA, complex passwords) increases security but hurts usability for frequent users. Mitigation: biometric login, trusted device tokens.
- Data minimisation: collecting less data improves privacy but may reduce personalisation quality (usability of features that depend on data).
- Audit trails: logging user actions for security creates privacy concerns. Mitigation: anonymisation, clear data retention policies.

13. Mobile vs Desktop Design

A recurring theme in recent papers (2022 q5, 2025 q5/q6) is adapting interfaces between desktop and mobile.

Aspect	Desktop	Mobile
Screen size	Large; can show much information simultaneously.	Small; requires prioritisation and progressive disclosure.
Input method	Mouse + keyboard; precise pointing; hover states.	Touch; finger is imprecise; no hover; fat-finger problem.
Navigation	Sidebar, top menu bars are accessible.	Bottom navigation bar or hamburger menu preferred (thumb zone).
Multitasking	Multiple windows; side-by-side apps.	Single app focus; context switching is more disruptive.
Context of use	Stationary, focused, adequate lighting.	On the move, one hand, variable lighting, interruptions.
Data entry	Keyboard; easy forms.	On-screen keyboard takes up half the screen; minimise typing.
Network	Usually stable broadband.	Variable; offline modes important; load times matter more.

13.1 Mobile Redesign Solutions (2022 paper)

Three aspects needing redesign with solutions:

- Navigation: sidebar nav → bottom navigation bar (within thumb reach; follows iOS/Android conventions).
- Data tables/dashboards: full-width table → card-based view with swipe gestures; or progressive disclosure accordion.
- Filter/search: complex dropdown filters → simplified single-criteria filter or search bar with autocomplete.

14. Evaluating Quantitative Findings (2020 Paper)

The 2020 exam presented three 'findings' and asked whether each supports a stated 'statement'. This tests critical thinking about data.

14.1 The Three Scenarios

Finding → Statement	Analysis
2/4 questionnaire respondents prefer not to use ring-back → 'Half of users do not use ring-back'	PARTIALLY supports but is weak. The finding says 'prefer not to use', not 'do not use'. Also, a sample of 4 is far too small to generalise to 'users' broadly. The statement overstates the finding.
One observer sees Joan walk 10 min to printer on one day → 'Significant time is wasted by designers'	Does NOT strongly support the statement. Single observation, single person, single day. Cannot generalise to 'designers' (plural) or to 'significant time'. Need longitudinal data across multiple designers.
Data log shows 8/1000 hours on helpfiles in Jan–Mar → 'Helpfiles used less than 1% of time in first quarter'	DOES support the statement. $8/1000 = 0.8\% < 1\%$, and January–March is exactly Q1. The calculation and temporal claim are accurate.

✓ Exam tip – Data analysis questions

- ✓ Always check: (1) does the finding literally say what the statement claims? (2) is the sample size adequate? (3) does the time period match? (4) is there a causal leap?
- ✓ State BOTH what is supported and what is NOT, with reasons.

15. Past Paper Topic Map (2019–2025)

Use this to identify high-frequency topics and gaps:

Year / Q	Q5 Topics	Q6 Topics
2019	UCD iterative cycle; Nielsen's heuristics (clothing website); Card sorting & similarity analysis	Primary stakeholder description; data gathering techniques; Gestalt theory with figures
2020	CUSTOM stakeholder analysis; Task analysis limitations; Evaluating quantitative findings; HE vs CW for visually impaired system	Primary stakeholders & task model (Weather App); Data collection for visually impaired; Gestalt theory (6 principles)
2021	Design a timepiece: functional/non-functional requirements; Lo-fi prototypes (sketches); HE of two prototypes	CW of weather app (5 tasks); Redesign suggestions from CW; Further data gathering after CW
2022	Gestalt (dashboard); CW instruction sheet (Find France sales); Mobile vs desktop redesign (3 aspects)	Data gathering methods (COVID app); Stakeholder & requirements analysis; Design principle; Usability vs security trade-off
2023	Human memory; Human attention; Information architecture; IA usability issue; Evaluation study design	User research methods (mindfulness app); Stakeholder analysis; Requirements; Design principle; Prototyping methods
2024	Stakeholder groups (ChatGPT); Heuristic Evaluation of ChatGPT; Interview questions for stakeholders; Interview data analysis	Requirements; Sketch design screens; User personas; Gestalt applied to design
2025	Requirements; Sketch app screens; Information architecture diagram; CW on designed app; Redesign from CW	HE violations → identify heuristics (Teams); Human attention (Teams); Human memory (Teams); Design principle; Adoption problem approach

High-frequency topics (appear most years):

- Nielsen's heuristics / Heuristic Evaluation: 2019, 2021, 2022, 2024, 2025
- Cognitive Walkthrough: 2021, 2022, 2025
- Gestalt theory: 2019, 2020, 2022, 2024
- Stakeholder analysis / requirements: every year
- UCD process / prototyping: 2019, 2020, 2021, 2023, 2025
- Data gathering methods: 2019, 2020, 2022, 2023, 2024

16. Exam Strategy & Mark-Scoring Tips

16.1 Reading the Question

- Read the mark allocation for each part – it tells you how much to write.
- 3 marks = roughly 3–4 substantive points. 8 marks = a detailed multi-point answer.
- 'Describe' means explain what something is. 'Discuss' means evaluate pros and cons. 'Identify' means name + brief justification.

16.2 Mark-Scoring Strategies by Question Type

Question type	How to score marks
List X things (heuristics, stakeholders, requirements)	Name each item clearly + give ONE sentence of explanation/justification. Marks are for the justification, not just the name.
Evaluate an interface (HE or CW)	For HE: name the heuristic, identify WHERE on the interface it is violated, explain WHY it's a violation. For CW: go action by action; answer the 4 questions; identify failures specifically.
Sketch a design	Draw clear labelled screens. Add captions explaining interactive elements. Show how it meets the stated requirements.
Compare two approaches (HE vs CW)	Give at least 3 criteria; for each, state the difference and which is better in THAT context. Don't just describe both methods.
Design principle	State the principle clearly. Motivate it with reference to the SPECIFIC user group and their needs. Link to data gathered or requirements.
Requirements	Functional + non-functional. State each as a requirement ('The app shall...'). Give a rationale for each.
Gestalt question	Name the principle → point to WHERE it appears in the given image → explain the EFFECT on the user. Three steps every time.

16.3 Common Mistakes to Avoid

- Describing a method without applying it to the specific context given.
- Listing heuristics without identifying actual violations in the shown interface.
- Vague requirements ('The app should be user-friendly') instead of testable ones.
- Generic personas (avoid 'John, 25, likes technology') – make them specific to the app's domain.
- Gestalt: naming a principle without showing where it's used in the specific interface.
- Forgetting to calculate similarity ratings step by step (show working).

16.4 Time Management in the Exam

Paper 3 typically allocates 40 minutes per question. With 20 marks per question:

- Allocate roughly 2 minutes per mark as a guide.
- Don't over-write on 3-mark sub-parts at the expense of 8-mark ones.
- Sketching questions: spend ~3–4 minutes drawing cleanly; labels are as important as the sketch.

17. Quick-Reference Cheat Sheet

Nielsen's 10 Heuristics – One-line summary

30. Visibility of system status – keep users informed.
31. Match system and real world – speak the user's language.
32. User control and freedom – easy undo/exit.
33. Consistency and standards – follow conventions.
34. Error prevention – stop errors before they happen.
35. Recognition not recall – make options visible.
36. Flexibility and efficiency – shortcuts for experts.
37. Aesthetic and minimalist design – no clutter.
38. Help recognise and recover from errors – plain error messages.
39. Help and documentation – easy searchable help.

6 Gestalt Principles

40. Proximity – close = related.
41. Similarity – looks-alike = same group.
42. Continuity – eye follows smooth paths.
43. Closure – perceive complete shapes from incomplete.
44. Figure/Ground – focus vs background.
45. Common Fate – moving together = related.

4 CW Questions

46. Will users try to do the right thing?
47. Will they notice the correct action is available?
48. Will they associate the action with the desired effect?
49. Will they see progress / feedback after the action?

CUSTOM Stakeholder Types

- C – Clients (commission/fund the system)
- U – Users (direct interaction)
- S – Secondary (affected, not direct users)
- T – Tertiary / Other (regulators, wider society)

Key Methods Summary

Method	One-line description
Heuristic Evaluation	Expert review against 10 heuristics; identifies broad usability violations.
Cognitive Walkthrough	Task-by-task learnability analysis answering 4 questions per action.
Interview	Conversational data gathering; qualitative, rich, small sample.

Method	One-line description
Questionnaire	Written survey; quantitative, large sample, less depth.
Observation	Watch users in context; reveals real behaviour vs stated preferences.
Card Sorting	Users sort cards into groups; informs IA.
Similarity Analysis	Calculates % agreement between users on card co-grouping.
HTA	Hierarchical decomposition of tasks into sub-tasks and actions.
Persona	Fictional user archetype derived from research; guides design decisions.
Lo-fi prototype	Cheap early design representation (paper sketches); fast to iterate.
Hi-fi prototype	Detailed digital representation; close to final product.

18. Practice Questions with Guidance

Q1: Heuristic Evaluation (high probability topic)

Practise evaluating ANY interface you use daily. For each screen, ask: which of the 10 heuristics are violated? Use this table for each violation:

Heuristic #	Location in interface	Why it is a violation
e.g. 5 – Error prevention	The 'Delete all chats' button (ChatGPT settings)	No confirmation dialog before irreversible deletion. Users could accidentally delete all history with no recovery option.
e.g. 1 – Visibility of status	After sending a message	No clear loading indicator while waiting for AI response on slow connections.

Q2: Cognitive Walkthrough practice

Choose a task in any app you use. Write the action sequence (HTA). Then for each action answer the 4 CW questions. Flag any failures and convert them to redesign suggestions.

Q3: Gestalt application

Open any website. Identify all six Gestalt principles in use. For each, note: where it appears, which principle, and what design effect it achieves. This trains you for the 'describe which principles are used in this figure' questions.

Q4: Sketch a design + requirements

Choose a hypothetical app: a meditation app for elderly users; a food-tracking app for diabetics; a revision tool for A-level students. Then: (a) write 3 requirements with rationale; (b) sketch 3 screens with labels; (c) identify 2 user personas; (d) note which Gestalt principles you're applying.

Q5: Data gathering method selection

For each scenario below, choose the most appropriate data gathering methods and justify:

50. Designing an app for stroke rehabilitation patients. (Cannot type, may have cognitive impairment.)
51. Designing a revision app for a large, diverse university. (Many users, quick turnaround needed.)
52. Redesigning a complex professional financial trading platform.

Practise justifying WHY a method is suitable AND identifying methods that would be DIFFICULT and why.

19. Glossary

Term	Definition
Affordance	A property of an object that suggests how it can be used (e.g. a button affords clicking).
Cognitive load	The mental effort required to use a system. High cognitive load leads to errors and dissatisfaction.
Conceptual design	High-level design of what a system will do and how users will interact with it conceptually.
Physical design	Detailed design of the interface look-and-feel, including layout, colour, and typography.
Ecological validity	The degree to which findings from a study reflect real-world conditions.
Fidelity	The degree to which a prototype resembles the final product. Lo-fi = rough; hi-fi = detailed.
Formative evaluation	Evaluation conducted during design to guide improvement.
Summative evaluation	Evaluation conducted after completion to assess overall quality.
Information architecture (IA)	Organisation and labelling structure of a system's content.
Mental model	A user's internal understanding of how a system works, based on prior experience.
Naturalistic observation	Observing users in their real, natural environment without interference.
Think-aloud protocol	Users verbalise their thoughts while completing tasks; reveals reasoning and confusion.
Triangulation	Using multiple data sources or methods to improve reliability and validity of findings.
Usability	The ease with which a specific user can use a system to achieve specific goals in a specific context.
User experience (UX)	The holistic experience of a person using a product, including usability, aesthetics, and emotion.
Wizard of Oz	Testing technique where a person simulates system responses to test interaction ideas without implementation.

Good luck in the Tripos!

Cambridge Computer Science Part IA – Interaction Design